REMARKS

Claims 1-15 and 17-21 are pending in the application.

Claims 1-15 and 17-21 have been rejected.

Reconsideration of the claims is respectfully requested.

I. CLAIM REJECTIONS -- 35 U.S.C. § 102

Claims 1-4, 6-10, 12-15, 17-19, and 21 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,740,178 to *Jacks et al.*, hereinafter "Jacks". This rejection is respectfully traversed.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. MPEP § 2131, p. 2100-76 (8th ed., rev. 4, October 2005) (*citing In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990)). Anticipation is only shown where each and every limitation of the claimed invention is found in a single prior art reference. *Id.* (*citing Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987)).

Claim 1 requires "a shadow memory initializer operable to detect an initialization event, to initialize a shadow memory based on the initialization event, and to calculate original verification data for the shadow memory, the shadow memory comprising shadow data".

The Examiner makes reference to Jacks col. 1, lines 35-61, col. 4, lines 13-17, and col. 5 lines 7-23, without attempting at all to identify which elements of Jacks are alleged to correspond to the claim elements, other than that Jacks' RAM is evidently believed to correspond to the claimed shadow memory. These passages together teach:

> The above problem is solved and an advance is made over prior art accordance with applicants' invention wherein the EEPROM back up memory stores a program for controlling the update of the RAM; this makes it unnecessary to rely on programs in the RAM to perform the updating function. Further the backup memory contains data which allows that data to be copied into different segments of the RAM, and contains hash sums which, when the RAM has been fully initialized, should reflect the state of initialized RAM. The EEPROM is much more expensive than RAM and is advantageously much smaller in total capacity than RAM; the RAM when used contains a great deal of dynamic data whereas the initialization data in the EEPROM is only required for programs and static data. Thus the RAM is not simply a copy of the contents of a part of the EEPROM but is a copy of blocks of EEPROM scattered in various locations in the RAM. Under these circumstances to insure the memory has been properly copied, the EEPROM stores a series of cyclic redundancy check (CRC) sums reflecting CRC sums generated for a fully initialized RAM. A check of the initialization process can therefore be made by generating these CRC sums over the contents of the initialized RAM and comparing them with the pre-stored CRC sums. *Col. 1, lines 35-61.*
>
> The EEPROM also contains additional data for use in initializing the RAM for subsequent initialization of the RAM of another (subtending) processor. The unpack information is used to locate this additional data. The initialized RAM contains a program for unpacking this additional data. *Col. 4, lines 13-17.*
>
> FIG. 5 illustrates the process of initializing the RAM from the EEPROM. The switch modular processor sends a signal to the common control to cause the common control to be reset (action block 502) which causes the common control to execute code stored in ROM (action block 504). Action block 506 is used to determine which half of the EEPROM is to be used for initializing. This is done by examining the first word, i.e., the word that contains the unique key and sequence number of each half of the EEPROM in order to determine whether either of these words is zero (indicating cleared memory), or if neither is zero, which sequence number and key represents the latest update. Since, following a successful update of one of the halves of the EEPROM, the other half is cleared, such a situation can only happen if the request to

> initialize RAM is received while an EEPROM half was being updated and the last step of the update had not been completed. Therefore, the half identified by the later (higher) sequence number accompanied by a valid key is the half that should be used for initializing the RAM. *Col. 5, lines 7-23.*

Nothing in these passages, or any other part of Jacks, appears to teach a "shadow memory initializer" (SMI) that is operable to perform all the functions required by claim 1. Nothing at all in Jacks is taught to detect an initialization event, though there is discussion of initializing the RAM. Nothing in Jacks therefore teaches an SMI that detects and initialization event and initializes the RAM based on the initialization event.

Further, nothing in Jacks is taught to be an SMI that also calculates original verification data for the shadow memory, as claimed. Jacks does state that the backup memory contains hash sums which, when the RAM has been fully initialized, should reflect the state of initialized RAM – but does not at all, ever, teach or suggest calculating these hash sums, and it appears these "hash sums" may be calculated from the EEPROM, not the RAM. In fact, col. 1, line 43 is the only reference at all in Jacks to a hash.

Jacks does mention "generating these CRC sums over the contents of the initialized RAM", but does not teach or suggest that this is performed by the same unit (as the claimed SMI) that

performs the other functions as claimed. For the analysis below, Applicant assumes that these "generated CRC sums" are alleged by the Examiner to be the claimed "original verification data" calculated by the SMI.

As such, Jacks does not teach anything identical to the SMI, as required by a proper anticipation rejection. Claim 6 includes similar limitations, and so claims 1-12 cannot be anticipated by Jacks.

Claim 1 also requires a shadow memory verifier (SMV) operable to detect a verification event and to verify the shadow data based on the verification event by calculating current verification data for the shadow memory and determining whether the current verification data matches the original verification data. In addition to the passages above, the Examiner cites col. 4, lines 27-41:

> Action block 404 corresponds to line 7 of FIG. 1. Test 406 is then used to determine whether the contents of the RAM are equivalent to the contents of the uncleared half of the EEPROM. The test is performed by deriving data that would be written into an EEPROM for each of the locations of the EEPROM and checking whether the derived data is equal to the data already stored in the EEPROM. (The key and sequence number stored in location 301 are not used by the switch module processor as an indication of the issue number of an update, since any mistakes in generating such a number would lead to unnecessary EEPROM erase cycles.) If it is found that the contents of the RAM are equivalent to the contents of the EEPROM memory, then the action is complete (action block 408) and effectively it is not necessary to update the EEPROM.

Jacks does not teach or suggest an SMV that is operable to perform all the functions required by claim 1. Nothing at all in Jacks is taught to detect a verification event and to verify the shadow data based on the verification event.

While Jacks does describe testing the contents of the RAM, Jacks does so by deriving data that would be written into an EEPROM for each of the locations of the EEPROM and checking whether the derived data is equal to the data already stored in the EEPROM. This is not the same as claimed, which would requires calculating current verification data for the shadow memory (RAM, according to the Examiner) and determining whether the current verification data matches the original verification data (the "generated CRC sums", according to the Examiner).

Note col. 4, lines 1-3 indicates that the "third unpack function" performs a CRC check on data copied into the RAM (interpreted by the Examiner as the "shadow memory") and compares this with information stored in the EEPROM sixth block as described above. Col. 3, lines 39-44 indicate that Jacks performs "a CRC check over all the blocks in segment 319 of the EEPROM memory. A CRC check sum is storm [*sic*] for every two kilobytes (2K) of this block of memory in a sixth segment of segment 319." This indicates that Jacks performs a comparison between a CRC check of the RAM with a CRC check of the EEPROM, not between current and original verification data calculated from the shadow memory, as claimed.

Jacks does not at any point appear to calculate current verification data for the shadow memory and determine whether the current verification data matches the original verification data, as claimed. Nor does Jacks teach a single unit, as the claimed SMV, that performs all these functions (nor a combined SMI/SMV).

As such, Jacks does not teach anything identical to the SMV, as required by a proper anticipation rejection. Claim 6 includes similar limitations, and so claims 1-12 cannot be anticipated by Jacks.

Claim 13 includes limitations similar to those above, without requiring specific units to perform various functions. Claim 13 requires verifying the shadow data based on the verification event by calculating current verification data for the shadow memory and determining whether the current verification data matches the original verification data.

As described above, Jacks teaches a comparison between a CRC check of the RAM with a CRC check of the EEPROM, not between current and original verification data calculated from the shadow memory, as claimed.

As such, Jacks cannot anticipate claims 13-15 and 17-21, either.

Accordingly, the Applicant respectfully requests the Examiner to withdraw the § 102 rejection with respect to these claims.

II. CLAIM REJECTIONS -- 35 U.S.C. § 103

Claims 5, 11, and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Jacks in view of Meyer, U.S. Patent 5,953, 352. The Applicant respectfully traverses the rejection.

In *ex parte* examination of patent applications, the Patent Office bears the burden of establishing a prima facie case of obviousness. MPEP § 2142, p. 2100-133 (8th ed. rev. 4, October 2005). Absent such a prima facie case, the applicant is under no obligation to produce evidence of

nonobviousness. *Id.* To establish a *prima facie* case of obviousness, three basic criteria must be met: *Id.* First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. *Id.* Second, there must be a reasonable expectation of success. *Id.* Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *Id.* The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *Id.*

Meyer also fails to teach or suggest the limitations described above that distinguish over Jacks. As such, no combination of Meyer and Jacks can teach or suggest the limitations of independent claims 1, 6, and 13. As such, all claims distinguish over any combination of the cited references.

Accordingly, the Applicant respectfully requests the Examiner to withdraw the § 103 rejection with respect to these claims.

CONCLUSION

As a result of the foregoing, the Applicant asserts that the remaining Claims in the Application are in condition for allowance, and respectfully requests an early allowance of such Claims.

If any issues arise, or if the Examiner has any suggestions for expediting allowance of this Application, the Applicant respectfully invites the Examiner to contact the undersigned at the telephone number indicated below or at ***wmunck@munckbutrus.com***.

The Commissioner is hereby authorized to charge any additional fees connected with this communication or credit any overpayment to Deposit Account No. 50-0208.

Respectfully submitted,

MUNCK BUTRUS P.C.

Date: Aug. 24, 2007

William A. Munck
Registration No. 39,308

P.O. Drawer 800889
Dallas, Texas 75380
(972) 628-3600 (main number)
(972) 628-3616 (fax)
E-mail: *wmunck@munckbutrus.com*